PERGAMON

Safety Science 40 (2002) 765–780

SAFETY SCIENCE

www.elsevier.com/locate/ssci

Software tools to support incident reporting in safety-critical systems

Chris Johnson*

Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK

Abstract

Incident reporting systems are playing an increasingly important role in the development and maintenance of safety-critical applications. The perceived success of the FAA's Aviation Safety Reporting System (ASRS) and the FDA's MedWatch has led to the establishment of similar national and international schemes. These enable individuals and groups to report their safety concerns in a confidential or anonymous manner. Unfortunately, many of these systems are becoming victims of their own success. The ASRS and MedWatch have both now received over 500,000 submissions. In consequence, the administrators of incident reporting systems increasingly rely upon software tools to support the administration of their systems. In the past, these systems have relied upon ad hoc applications of conventional database technology. However, there are several reasons why this technology is inadequate for many large-scale reporting schemes. In particular, the problems of query formation often result in poor precision and recall. This, in turn, has profound implications for safety-critical applications. Users may fail to identify similar incidents within national or international collections. These ad hoc approaches also neglect the opportunities provided by recent developments in computer assisted interviewing and in the monitoring of retrieval activities to build models of user behavior. These techniques offer a number of potential benefits. For instance, it is possible to automatically detect potential biases in the way that investigators analyze particular incidents.

Keywords: Case based reasoning; Information retrieval; Incident reporting; Accident reporting

1. Introduction

Incident reporting schemes are increasingly being seen as a means of detecting and responding to failures before they develop into major accidents. For instance, part of the UK government's response to the Ladbroke Grove crash has been to establish

* Tel.: +44-141-330-6053; fax: +44-141-330-4913.
E-mail address: johnson@dcs.glasgow.ac.uk (C. Johnson).

0925-7535/02/$ - see front matter
PII: S0925-7535(01)00085-6

a national incident-reporting scheme for the UK railways. At a European level, organizations such as Eurocontrol have been given the responsibility of establishing international standards for the reporting schemes that are operated by member states. In the United States, the Senate recently set up the Chemical Safety and Hazard Investigation Board to coordinate incident reporting throughout the chemical industries. The popularity of these schemes depends upon their ability to elicit reports from operators. This, in turn, depends upon individuals receiving the feedback that is necessary to demonstrate that their participation is both valued and worthwhile. People will only submit if they believe that their contributions will be acted on. In part this depends upon the confidentiality of the system. Individuals must not fear retribution providing that they are not reporting criminal activity. However, an effective response and individual participation also rely upon our ability to analyze and interpret the submissions that are made to incident reporting schemes.

In the past, most incident reporting schemes in safety-critical industries have operated at a local level. For instance, chemical and steel companies have developed proprietary systems that operate within their plants (van Vuuren, 1998). In the UK health service, this has led to a situation where there are many different local schemes with no effective means of sharing data between hospitals. This situation is not as pathological as it might appear. Local schemes have the benefit that individual contributors can directly monitor the impact of their contributions on their working environment. The people maintaining these systems can also inspect the systems and environments in which incidents occur (Busse and Johnson, 1999). However, the disadvantages are equally apparent. There is a danger that the lessons learnt in one institution will not be transferred to other organizations. There is also a danger that individual incidents may appear as isolated instances of failure unless there is confirmatory evidence of similar incidents occurring on a national and international scale (Dunlop et al., 1998). For all of these reasons there is now an increasing move towards national and international systems. Later sections will describe how this is introducing new problems of scale that can only be solved with software support.

The US Food and Drug's Administrations MedWatch program provides an example of a nationwide incident reporting system. This scheme enables healthcare professionals to report incidents involving medical devices. They are asked for rudimentary details about particular incidents. This is justified because the system is confidential and not anonymous. The operators of the MedWatch programme can, therefore, contact the respondents to elicit further information. Fig. 1 presents two MedWatch reports that deal specifically with software "failures" in medical devices. They are typical of the information that is lodged with incident reporting schemes. A number of categories are used to help index the data and to support subsequent statistical analysis. In the case of the MedWatch programme, this includes information about the particular types of devices that were involved, the Product Code. The classification information also includes the clinical area that the device was being used in, the Panel Code. Free text is also used to provide details about how the incident was detected and was resolved, the Event Description.

Access Number: MXXXXXX	Date Received: 12/13/93
Manufacturer Code: XXXXXXX	Report Type: MALFUNCTION
Catalog Number: NI	Product Code: JJC
Panel Code: CLINICAL CHEMISTRY	Event Type: FINAL

Event Description: clinical chemistry analyzer erroneously printed out a value of >5 as the result of a lipase control, but transmitted the correct value of 39 to the laboratory's host computer. The software for the analyzer was evaluated by its mfr and was found to contain a software bug which caused the inappropriate printing of a qualitative parameter when the laboratory host computer and data printing accessed the qualitative data processing program at the same time. Software mfr has modified the software, and an evaluation of the revised software is in progress at this co. (*)

Access Number: MXXXXXX	Date Received: 09/25/95
Manufacturer Code: XXXXXXX	Report Type: MALFUNCTION
Catalog Number: NA	Product Code: LWS
Panel Code: CARDIOVASCULAR	Event Type: FINAL

Event Description: co received info from a field clinical engineer that while using software module during a demonstration with a non functional implantable cardioverter defibrillator, noted that two of the keys were mapped incorrectly. The field clinical engineer then changed to another software module, same model, serial number 006000. And experienced the same results. Corrective action: co has recalled and requested the return of affected revision 11.2 software modules and replaced them with the appropriate version. (*)

Fig. 1. Examples of software failure from the FDA's MedWatch programme.

2. The Problems

Incident reporting systems have become victims of their own success. The FAA has maintained a consistently high participation rate in the ASRS since it was established in 1976. It now receives an average of more than 2600 reports per month. The cumulative total is now approaching half a million reports. MedWatch was set up by the FDA as part of the Medical Devices Reporting Program in 1984. It now contains over 700,000 reports. These figures are relatively small when compared with the size of other data sets that are routinely maintained in many different industries. However, the safety-critical nature of these reports creates a number of unique problems that frustrate the development of appropriate software support.

2.1. Elicitation

It can be difficult to elicit information about previous incidents from the users that are involved in adverse incidents. Many reporting forms provide a cursory overview

of the events leading to failure and so investigators have to visit contributors to identify missing information. This creates considerable logistical problems for the growing numbers of national and international reporting systems. As more reports are received then increasing numbers of investigators may be required to conduct follow-up interviews. If additional staff are not provided then a significant period of time can elapse between a report being submitted and the subsequent elicitation of additional details. During this period, it is increasingly likely that witnesses will forget significant details. There is also a danger that implicit and explicit pressures may influence their account of a particular failure. This point can be illustrated by the biases that affect eyewitness testimony:

- Confidence bias. This arises when witnesses unwittingly place the greatest store in their colleagues who express the greatest confidence in their view of an incident. Previous work into eye-witness testimonies and expert judgements has shown that it may be better to place greatest trust in those who do not exhibit this form of over-confidence (Johnson, in press).
- Hindsight bias. This form of bias arises when witnesses criticize individuals and groups on the basis of information that may not have been available at the time of an incident.
- Judgement bias. This form of bias arises when witnesses perceive the need to reach conclusions about the cause of an incident. The 'quality' of the analysis is less important that the need to make a decision.
- Political bias. This arises when a judgement or hypothesis from a high status member commands influence because others respect that status rather than the value of the judgement itself. This can be paraphrased as 'pressure from above'.
- Sponsor bias. This form of bias arises when a witness testimony can indirectly affect the prosperity or reputation of the organization that they manage or are responsible for. This can be paraphrased as 'pressure from below'.
- Professional bias. This arises when witnesses may be excluded from the society of their colleagues if they submit a report. This can be paraphrased as 'pressure from beside'.
- Recognition bias. This form of bias arises when witnesses have a limited vocabulary of causal factors. They actively attempt to make any incident 'fit' with one of those factors irrespective of the complexity of the circumstances that characterize the incident.
- Confirmation bias. This arises when witnesses attempt to make their evidence confirm an initial hypothesis.
- Frequency bias. This form of bias occurs when witnesses become familiar with particular causal factors because they are observed most often. Any subsequent incident is, therefore, likely to be classified according to one of these common categories irrespective of whether an incident is actually caused by those factors.
- Recency bias. This form of bias occurs when a witness is heavily influenced by previous incidents.

- Weapon bias. This form of bias occurs when witnesses become fixated on the more 'sensational' causes of an incident. For example, they may focus on the driver behavior that led to a collision rather than the failure of a safety-belt to prevent injury to the driver.

It is unlikely that any software system will be able to entirely eliminate all of these different forms of bias. As we shall see, however, software can be used to reduce the delay between an incident being reported and the elicitation of additional contextual information. It is also possible to perform increasingly complex linguistic analyses that can help to identify different forms of bias. For example, anxiety bias stems from two different factors. State anxiety is a natural reaction to stressful situations that investigators can anticipate would affect everyone involved in an incident. In contrast, trait anxiety affects particular people who are naturally more anxious than the rest of the population irrespective of their particular circumstances. These individuals can be detected by their aversion to using particular terms such as 'death' or 'injury'. Trait anxiety is also, typically, indicative of poor medium-term recall in the aftermath of high-stress situations.

2.2. Precision and Recall

A number of further problems complicate the retrieval of incident reports once they have been submitted to large-scale reporting systems. Precision and recall are concepts that are used to assess the performance of all information retrieval systems. In broad terms, the precision of a query is measured by the proportion of all documents that were returned which the user considered to be relevant to their request to the total number of documents that were returned. In contrast, the recall of a query is given by the proportion of all relevant documents that were returned to the total number of relevant documents in the collection (Dunlop et al., 1998). It, therefore, follows that some systems can obtain high recall values but relatively low precision. In this scenario, large numbers of relevant documents will be retrieved together with large numbers of irrelevant documents. This creates problems because the user must then filter these irrelevant hits from the documents that were returned by their initial request. Conversely, other systems provide high precision but poor recall. In this situation, only relevant documents will be returned but many other potential targets will not be retrieved for the user.

In most other areas of software engineering, the trade-off between precision and recall can be characterized as either performance or usability issues. In incident reporting schemes, these characteristics have considerable safety implications. For instance, low-recall systems result in analysts failing to identify potentially similar incidents. This entirely defeats the purpose of compiling national and international collections. More worryingly in a commercial setting it leaves companies open to litigation in the aftermath of an accident. Failure to detect trend information in previous incident reports can be interpreted as negligence. Conversely, low-precision approaches leave the analyst with an increasing manual burden as they are forced to continually navigate "another 10 hits" to slowly identify relevant reports from those

that have no relation to their information needs. Again this can result in users failing to accurately identify previous records of similar incidents.

2.3. Data abstractions and dynamic classifications

A number of further problems complicate the software engineering of tool support for incident reporting systems. In particular, incidents will change over time. The introduction of new technology and working practices creates the potential for different forms of hardware and software failure as well as different opportunities for operator "error". Any data abstractions that are used to represent attributes of incident reports must also be flexible enough to reflect these changes in incident classification schemes. This problem arises because the incident classification schemes that regulators use to monitor the distribution of events between particular causal categories are, typically, also embodied in the data abstractions of any underlying tool support.

There are two general approaches to the problems of developing appropriate data models for incident reports. The first relies upon the use of generic categories. These include "software failure" rather than "floating point exception" or "human error" rather than "poor situation awareness". These high-level distinctions are unlikely to be extended and refined over time. However, they also result in systems that yield very low precision. A query about "floating point exceptions" will fail if all relevant reports are classified as "software failures". Further problems arise if inheritance mechanisms are used to refine these high level distinctions. The addition of new subtypes, for instance by deriving "floating-point exceptions" from "software failures", forces the reclassification of thousands of existing reports.

The second approach that addresses the changing nature of many incidents is to develop a classification scheme that is so detailed, it should cover every possible adverse event that might be reported. To provide an illustration of the scale of this task, the US National Co-ordinating Council for Medication Error Reporting and Prevention produces a Taxonomy of Medication Errors. This contains approximately 400 different terms that record various aspects of adverse incidents. EUROCONTROL have developed a similar taxonomy for the classification of human "error" in incident reports. There is no such taxonomy for software related failures. This is a significant issue because retrieval systems must recognise similar classes of failures in spite of the different synonyms, euphemisms and colloquialisms that are provided in initial reports of "bugs", "crashes", "exceptions" and "run-time failures". There are further more general problems. In particular, if safety-critical industries accept detailed taxonomies then software tools may exhibit relatively poor recall in response to individual requests. The reason for this is that many existing classification systems are exclusive. As can be seen from Fig. 1, incidents tend to be classified by single descriptors rather than combinations of terms. As a result, many incidents that stem from multiple systemic failures cannot easily be identified. There is also the related problem that national and international systems must rely on teams of people to perform the analysis and classification. This introduces problems of inter-classifier reliability. Systems that are based on a detailed

taxonomy increase the potential for confusion and ultimately low recall because different classifiers may exhibit subtle differences in the ways in which they distinguish between the terms in the taxonomy.

2.4. Inter-analyst Reliability

The biases that were listed in Section 2.1 not only affect an eye-witnesses account of an incident or accident. They also affect the analysts' view of the causes of any failure. There is a danger that rather than learning the lessons of the past, organizations will simply use incident reports to find evidence that supports their existing preconceptions and biases (Johnson, 2000b). For instance, Lekberg's (1997) work for the Swedish Nuclear Power Inspectorate illustrates the problems that arise in analyzing the contributions to incident reporting systems. She demonstrates that there are fundamental biases in the way that different experts analyze particular incidents. Previous training and expertise affect an engineer's interpretation of causal events. Individuals who have received previous training in human factors are more likely to diagnose human factors issues in an incident report that their colleagues who have not received this form of training. This finding has significant implications because inter-analyst biases can have a knock-on effect on the conclusions that are drawn about particular incidents. This, in turn, will affect the lessons that are drawn from previous failures.

One means of addressing these biases is to ensure that all analysts are trained to the same standard. Unfortunately, this has proven to be impossible for most large-scale incident reporting systems. Few organizations have the resources to ensure that all of their investigators attend anything but the most cursory of foundation courses. This contrasts sharply with accident investigation where bodies such as the NTSB have established colleges to train their employees. Further factors increase the diversity of backgrounds that can bias investigators' interpretation of safety-critical incidents. In particular, diversity is often seen as an important strength of many investigation agencies. This argument holds for small-scale systems when other analysts can account for the potential biases of their colleagues. However, as the scale of the system grows there may be less assurance that the findings of an investigation say more about an incident than they do about the investigator who drafted them.

3. Solutions: computer assisted interviewing

The problems of eliciting information about previous incidents should not be underestimated. At present many systems rely upon confidential rather than anonymous reporting. As mentioned, the MedWatch system exploits this approach. Similarly, NASA personnel go back to the contributors of many ASRS submissions. This requires considerable resources. There must be enough trained analysts to elicit the necessary information during follow-up visits. Alternatively, it might be possible to recruit novel computational techniques to improve the quality of information that

is initially contributed in response to an incident. These techniques can reduce the expense associated with site visits. Equally importantly, they might also avoid the biases that affect follow-up interviews. A number of social concerns must affect contributors during safety-related discussions with external interviewers. Eliciting more information in the immediate aftermath of an incident also helps to reduce any delay between the contribution of a report and a follow-up interview.

The problems of extracting information from domain experts has been addressed by work on knowledge elicitation in general and by computer-aided interviewing techniques in particular (Saris, 1991). These interviewing techniques, typically, rely upon frames and scripts that are selected in response to information from the user. For example, the user of an air traffic management system might first be prompted to provide information about the stage of flight in which an incident occurred. If it happened during landing then a script associated with that stage of flight would be selected. This might provide further prompts about the activities of arrivals and departures officers or about specific items of equipment, such as MSAW protection. These detailed questions would not be appropriate for incidents during other stages of flight, such as those filed during en route operations.

The relatively simple script-based techniques, described above, offer a number of further benefits. In particular, the use of computer assisted interviewing can reduce the biases that stem from the different approaches that are used by many interviewers. Inter-analyst reliability is a continuing concern in many incident report systems (Johnson, 2000b). Computer-based interviewing techniques can be used to ensure that particular questions are *always* asked in particular situations. The use of scripts and frames encourages this approach although great care is required to ensure that any particular dialogue is appropriate for the context in which it is delivered. The scripts embodied in computer assisted interviewing systems can also be tailored to elicit particular information about regulatory concerns. For instance, if previous accidents had indicated growing problems with workload distribution during certain team-based activities then scripts could be devised to specifically elicit information about these potential problems.

These advantages must be balanced against the obvious limitations of computer-based interviewing techniques (Saris, 1991). Our initial experience in applying these techniques within hospital-based incident reporting systems has shown that it can be difficult to tailor the dialogue to match the users expertise and experience in both using the reporting system and in recognizing the symptoms of the failure that they have observed. For example, if a nurse has observed a failure in a patient monitoring system it has proven to be difficult, if not impossible, to prompt them to provide further information beyond the immediate report that the system has failed. In other situations this technique has proved to be far more successful, especially if the system provides feedback about the likely time that support staff will rectify any fault. Further evidence is also needed to determine whether the weaknesses of computers assisted interviewing in employment selection or the analysis of consumer behavior also apply to their application in incident reporting. Until this evidence is provided then there will continue to be significant concerns about the problems of bias that can be introduced during the elicitation of information about previous failures.

4. Solutions: relational data bases

There are two central tasks that users wish to perform with large-scale incident reporting systems. These two tasks are almost contradictory in terms of the software requirements that they impose. On the one hand, there is a managerial and regulatory need to produce statistics that provide an overview of how certain types of failures are reduced in response to their actions. On the other hand, there is a more general requirement to identify trends that should be addressed by those actions in the first place. The extraction of statistical information typically relies upon highly-typed data so that each incident can be classified as unambiguously belonging to particular categories, such as those described in the previous section. In contrast, the more analytical uses of incident reporting systems involve people being able to explore alternative hypotheses about the underlying causes of many failures. This, in turn, depends upon less directed forms of search. Unfortunately, most incident reporting systems seem to be focussed on the former approach. Relatively, few support these more open analytical activities.

Many incident reporting systems exploit relational database techniques. They store each incident as a record. Incident identifiers, such as the classified fields before the free text descriptions in Fig. 1, are used to link, or join, similar records in response to users' queries. It is important to emphasize that many existing applications of this relational technology have significant limitations. They are, typically, built in an ad hoc manner using mass-market database management systems. The results are often very depressing. For example, Boeing currently receives data about maintenance incidents from many customer organizations. Each of these organizations exploits a different model for the records in their relational systems. As a result, the aircraft manufacturer must attempt to unify these ad hoc models into a coherent database. At present, it can be difficult or impossible for them to distinguish whether a bolt has failed through a design fault or through over torquing by maintenance engineers. Sam Lainoff recently summarized the problems of populating their relational database in the following way:

> There is no uniform reporting language amongst the airlines, so it's not unusual to find ten different ways of referring to the same thing. This often makes the searching and sorting task a difficult proposition...The data we have won't usually permit us to create more refinement in our error typing. But at times it will give us enough clues to separate quality problems, and real human error from pure hardware faults. (Lainoff, 1999).

This quotation illustrates a couple of points. Firstly, it identified the commercial importance of these problems within safety-critical industries. Secondly, it is indicative of the problems that people face when attempting to correctly assign values to the fields that are defined in relational databases. This problem stems from the diverse and changing nature of incident reports that was described earlier. However, this quotation does not reveal all of the problems that are created by relational approaches. In particular, it can be extremely difficult for people who were not

involved in the coding and classification process to develop appropriate queries. One example query in a relational incident reporting system within the steel industry was expressed as follows:

```
$ SEL 1; USE EMP; INDEX SEV TO T1; SEL 2; USE DEPT; INDEX SEV
TO T2; SET REL EMP SEV; DISP NAME SEV ID DATE
```

even professional software engineers fail to retrieve correctly indexed records using relational query languages such as SQL (Reimers and Chung, 1993). These findings are not significantly effected even when graphical and menu-driven alternatives are provided.

5. Solutions: free-text retrieval and probabilistic inference

Information retrieval tools provide powerful mechanisms for indexing and searching large collections of unstructured data. They have supported numerous applications and are ubiquitous on the World Wide Web. It is, therefore, surprising that they have not been more widely adopted to support incident reporting systems. One explanation for this is that they cannot, in their pure form, be used to collate the statistics that are more easily extracted using relational systems. However, they avoid many of the problems associated with database query languages. In particular, they offer a range of techniques for exploiting semantic information about the relationships between the terms/phrases that appear in a document and the terms/phrases that appear in the users' query. These techniques enable analysts to ensure that queries that include concepts such as "software failure" will also be associated with terms such as "Floating point exception" or "Null pointer error".

Information retrieval systems, typically, perform several indexing processes on a data set before it can be searched (Turtle and Croft, 1991). For instance, variations on Porter's stemming algorithm can be used to unify terms such as "failure", "failing" and "failed". This preliminary analysis also includes the compilation of dictionaries that support query expansion. For example, "Numeric Error Exception" and "Floating Point Exception" occur in similar contexts but are not synonyms. As a result, they may not be grouped within standard thesauri. Programmers and analysts can, however, provide this semantic information so that a retrieval engine will locate both forms of incident in response to a user's query about numeric software failures. These rather simplistic techniques are supported by more complex concept recognition. Information retrieval tools can exploit probabilistic information based on the relative frequencies of key terms (Turtle and Croft, 1991). The system can rank documents according to whether or not it believes that documents are relevant to a query. If a term such as "floating point exception" occurs in a query but is only used infrequently in the collection then those documents that do contain the term are assigned a relatively high probability of matching the query. This process of assigning probabilities can be taken one stage further by supporting relevance feedback. In this process, the user is asked to indicate which of the documents that the

system proposed were actually relevant to their query. The probabilities associated with terms that occur amongst several of the documents that are selected can then be increased.

Fig. 2 illustrates how the FDA have recently exploited some of the techniques mentioned earlier in their medical devices reporting system. As can be seen, this system also retains the ability to exploit the fields that were encoded in earlier relational approaches mentioned in the previous section. Unfortunately, these approaches still have a number of disadvantages when providing software support for incident reporting schemes. In particular, it is still difficult to tune queries in retrieval engines and in relational databases to improve both the precision and recall of particular searches. As a result, it is entirely possible for users to issue queries that fail to find similar incidents or which return almost every report in a collection of well over half a million incidents. We have recently conducted a number of tests to support this argument. We began by manually tagging any incident reports that dealt with a loss of separation in the last 100 ASRS Air Traffic Control submissions. These nine tagged reports provided a base case that enables us to judge whether the

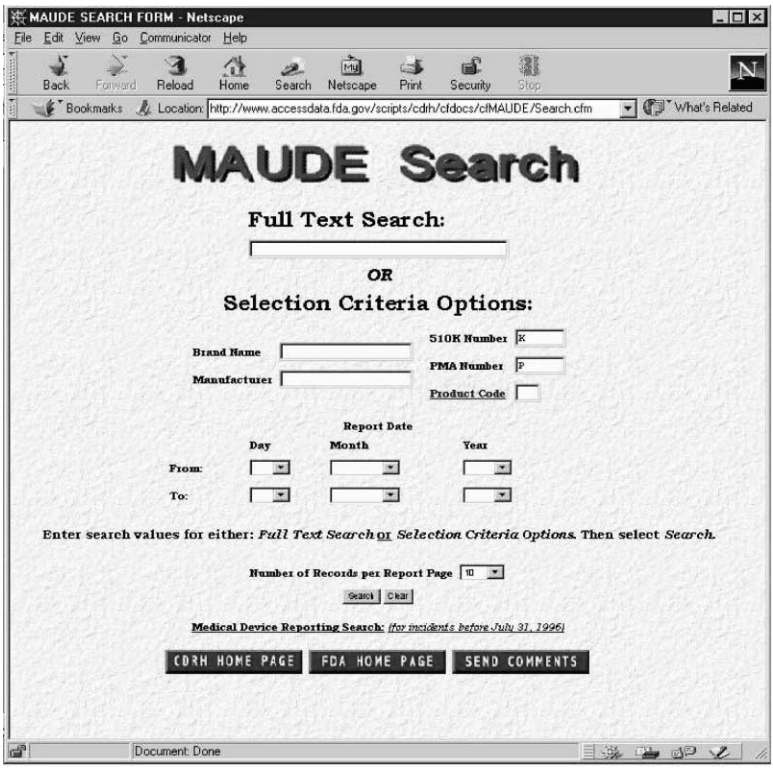

Fig. 2. Integrating information retrieval and relational techniques http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfMAUDE/Search.cfm.

retrieval engine performed as well as manual retrieval. We then indexed the same set of reports using the Inquery (Turtle and Croft, 1991) search engine and issued a query using the phrase "Aircraft loss of separation". As mentioned, recall is defined to be the number of relevant items retrieved divided by the number of relevant items in the database. In the first of our tests, all nine relevant reports were retrieved giving a maximum recall value of one. However, precision is defined to be the number of relevant items retrieved divided by the total number of items retrieved. Our query yielded 46 possible hits giving a precision of 0.19. In practical terms this meant that any investigator would manually have to search through the 46 potential hits to identify the nine relevant reports. This relatively poor precision can be improved by refining the query or by improving the internal weightings that Inquery uses for key terms, such as Aircraft, that may have biased the results of our query (Turtle and Croft, 1991; McElroy, 2000). There are, however, alternative means of providing software support for incident reporting systems.

6. Solutions: conversational search and CBR

Case-based reasoning (CBR) offers a further alternative to information retrieval techniques and relational databases. In particular, conversational case based reasoning offers considerable support for the retrieval of incident reports within safety-critical industries. For instance, the US Naval Research Laboratory's Conversational Decision Aids Environment (NaCoDAE) presents its users with a number of questions that must be answered in order to obtain information about previous hardware failures (Aha et al., 2001). For instance, if a user inputs the fact that they are facing a power failure then this will direct the system to assign greater relevance to those situations in which power was also unavailable. As a result, the system tailors the questions that are presented to the user to reflect those that can most effectively be used to discriminate between situations in which the power has failed. NaCoDAE was initially developed to support fault-finding tasks in non-safety critical equipment such as printers. We have recently extended the application of this tool to help analysts perform information retrieval tasks in large-scale incident reporting systems, including the FAA's ASRS. Fig. 3 illustrates this application of the NaCoDAE tool. After loading the relevant case library, the user types in a free-text query into the "Description" field. This is then matched against the cases in the library. Each case is composed of a problem description, some associated questions and if appropriate a description of remedial actions. The system then provides the user with two lists. The first provides "Ranked Questions" that the system believes are related to the user's original question. This helps to reduce the query formation problems that have been noted for other systems. The second "Ranked cases" list provides information about those cases that the system currently believes to match the situation that the user is confronted with. A particular benefit of this approach is that stratified case-based reasoning algorithms can be used to ensure that questions are posed in a certain order. They can help to ensure that users move from general questions that partition the case base at a gross level to increasingly

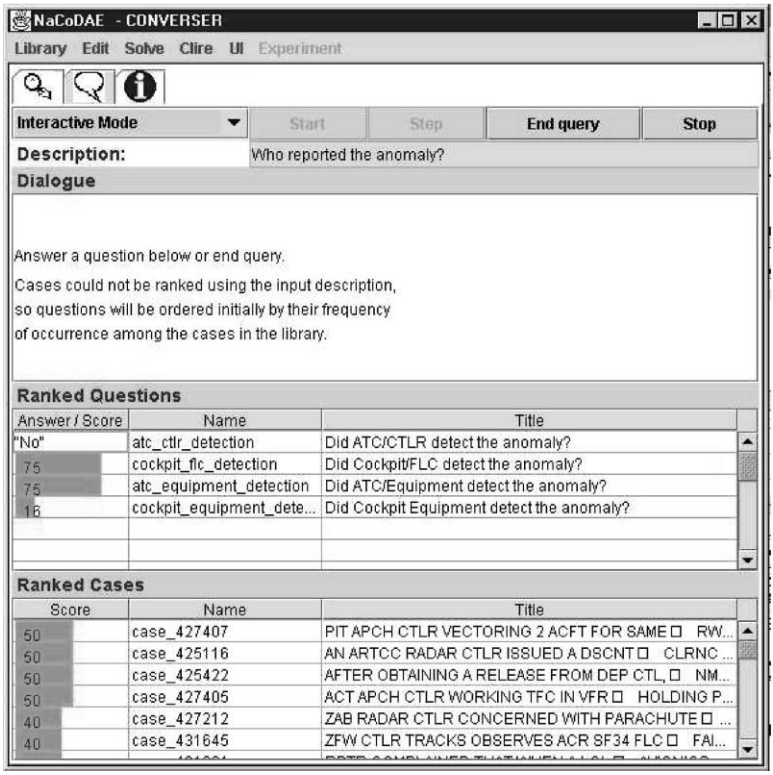

Fig. 3. Using NaCoDAE's conversational case-based reasoning (CBR) to support incident reporting.

precise questions that may only yield specific cases in response to their interactive search (Aha et al., 2001).

The previous paragraph indicates the critical nature of the questions that are encoded within the NaCoDAE system. Our study began by deriving these questions directly from the fields that are encoded in the MedWatch and ASRS systems. Users navigate the case base by answering questions about how the incident was resolved, what the consequences of the anomaly were, who identified the anomaly etc. If the user selected Cockpit/FLC as an answer to the question "Who detected the incident?" then all cases in which the flight crew did not detect the incident would be automatically moved down the list of potential matches. Each incident report only contains answers to some of these questions. For instance, the person submitting the form may not know how it was resolved. Once a set of similar cases has been identified, it can look for questions that can discriminate between those cases. For example, if some highly ranked cases were resolved by the Aircrew and others were resolved by Air Traffic Controllers then the system will automatically prompt the user to specify which of these groups they are interested in. This iterative selection of cases and prompting for answers from the user avoids the undirected and

often fruitless query formation that is a common feature of other approaches (Aha et al., 2001).

7. Solutions: tracing investigators' patterns of search

Section 2.4 summarized the problems of inter-analyst bias that affect the findings that can be drawn from many incident reports. These problems can be approached in a number of ways. For example, classification rules can be used to guide a causal analysis. These rules can be based on mathematical models of causation (Ladkin, 2000). Investigators can be guided by less formal heuristics as they select a small number of root causes from an approved taxonomy. Individual biases are reduced because the analysis procedures force analysts to explicitly consider a wide range of latent and catalytic factors, including human error, system failure, managerial weakness, environmental factors, etc. This latter approach has been exploited by EUROCONTROL to help standardize the analysis of human factors failures within many diverse national Air Traffic Management systems. They have developed a number of computer-based tools that guide the investigator towards a particular finding by asking them a number of questions about an incident. This is an extension of the computer-based interviewing techniques that exploit similar scripts when eliciting eyewitness statements. In this case, however, the dialogue is intended to ensure that all investigators will arrive at the same causal classification for similar incidents.

Unfortunately, this script-based approach suffers from the problems of static classification schemes mentioned in previous sections. If new causes are identified then analysis procedures must be revised and the software tools must be re-written. This need not be a significant problem unless the reporting system relies upon relational database technology. If this were the case then the revised dialogue would lead the investigator to identify a different set of causal factors. In order to preserve the consistency of their analysis, investigators would have to go back and reclassify every record in their data-set to ensure that it reflected the new analysis protocols.

We have begun to explore another technique that can be used to avoid some of the problems that can arise from the differences that exist between different investigators' interpretation of the same incident. These techniques rely upon the investigators' use of the probabilistic and case-based retrieval techniques that have been introduced in previous sections. As they interact with these systems, it is possible to maintain a log of their search patterns. These can then be analyzed to determine whether their retrieval of particular incidents can yield insights into potential biases. For example, we have recorded traces where individuals have navigated previous incidents purely in terms of the technical causes of a failure. Other individuals begin their searches by deliberately excluding all incidents that are classified as being caused by human factors issues. It is difficult to reach firm conclusions about the insights that these patterns yield about potential biases and so considerable further work is required. It is remarkable, however, that even rudimentary patterns can be used to identify

individual investigators as they search from previous examples of safety-critical failures (Johnson, in press).

8. Conclusions and Further Work

This paper stresses the growing importance that incident reporting systems have in many safety-critical industries. Unfortunately, many of these schemes currently rely on ad hoc implementations running on relational databases (Lainoff, 1999). These systems suffer from a number of problems. Poor precision and low recall may be dismissed as usability issues in other contexts. For safety-critical applications they may prevent analysts from identifying common causes both to software related incidents and other forms of failure. These problems are compounded by the difficulty of query formation in relational systems and by the problems of developing appropriate data models that reflect the ways in which incident reports will change over time. In contrast, information retrieval tools relax many of the problems that frustrate query formation in relational databases but they also make it difficult for users to assess the effectiveness of "naïve" queries. By "naïve" we mean that users may have no understanding of the probabilistic algorithms that determine the precision and recall of their query. We have proposed conversational case-based reasoning as a means of avoiding these limitations. This approach uses a combination of free-text retrieval techniques together with pre-coded questions to guide a user's search through increasingly large sets of incident reports. The application of tools, such as the US Navy's Conversational Decision Aids Environment, can be extended from fault finding tasks to support the retrieval of more general accounts of systems failure, human 'error' and managerial 'weakness'.

There are many alternative software-engineering techniques that can be applied to support national and international incident reporting systems. For example, our experience of information retrieval engines is largely based around extensions to Bruce Croft's Inquery tool (Turtle and Croft, 1991). The point of this paper is not, therefore, to advocate the specific algorithms that we have implemented or the systems that we have applied. It is, in contrast, to encourage a greater participation amongst software engineers in the design and maintenance of incident reporting software. If this is not achieved then the world's leading aircraft manufacturers will continue to have considerable difficulty in searching the incident data that is provided by their customers (Lainoff, 1999). If this is not achieved then there will continue to be medical reporting tools that fail to return information about incidents that users know have already been entered into the system (Johnson, 2000a).

We have also argued that computer-based interviewing techniques and the analysis of retrieval logs can be used to address the problems of bias that affect both eyewitness testimonies and expert analysis. These applications are more tentative; many fundamental problems remain to be addressed. For example, our initial experience has shown that some users are unwilling to enter into computer-generated dialogues about the incidents that they have witnessed. Similarly, it is far from clear whether the inferences that can be drawn from the retrieval logs of

investigators actually provide meaningful insights into their analysis of particular classes of incident.

References

Aha, D., Breslow, L.A., Munoz-Avila, H., 2001. Conversational case-based reasoning. Journal of Artificial Intelligence (in press).

Busse, D., Johnson, C.W., 1999. Human error in an intensive care unit: a cognitive analysis of critical incidents. In: Dixon, J. (Eds.), Seventh International Systems Safety Conference, Orlando, FL. International Systems Safety Society, Unionville, VA, USA, pp. 138–147.

Dunlop, M.D., Johnson, C.W., Reid, J., 1998. Exposing the layers of information retrieval evaluation. Interacting with Computers 10 (3), 225–237.

Johnson, C.W., 2000a Using case-based reasoning to support the indexing and retrieval of incident reports. In: Cottam, M.P., Harvey, D.W., Pape, R.P., Tait, J. (Eds.), Proceedings of European Safety and Reliability Conference Edinburgh (ESREL 2000): Foresight and Precaution. Balkema, Rotterdam, The Netherlands, pp. 1387–1394.

Johnson, C.W., 2000b. Using incident reporting to combat human error. In: Cockton, G. (Ed.), People and Computers XIV: Proceedings of HCI 2000. Springer Verlag, Berlin.

Johnson, C.W. Incident reporting: a guide to the detection, mitigation and resolution of failure in safety-critical systems (in press, to be published Spring 2002, Springer Verlag).

Ladkin, P.B., 2000. Causal reasoning about accidents. In: Koorneef, F., van der Meulen, M. (Eds.), SAFECOMP 2000. Springer Verlag, Berlin, Germany, pp. 344–355. (Lecture Notes in Computing Science No. 1943).

Lainoff, S., 1999. Finding human error evidence in ordinary airline event data. In: Koch, M., Dixon, J. (Eds.), Seventh International Systems Safety Conference Seattle, WA. International Systems Safety Society, Unionville, VA, USA.

Lekberg, A., 1997. Different approaches to accident investigation: how the analyst makes the difference. In: Moriarty, B. (Ed.) Proceedings of the Fifteenth International Systems Safety Conference. International Systems Safety Society, Sterling, VA.

McElroy, P., 2000. Information Retrieval/Case-Based Reasoning for Critical Incident and Accident Data. Project Dissertation, Department of Computing Science, University of Glasgow, Scotland..

Reimers, P.E., Chung, S.M., 1993. Intelligent user interface for very large relational databases. Proceedings of the Fifth International Conference on Human–Computer Interaction 2, 134–139.

Saris, W.E., 1991. Computer Assisted Interviewing. Sage, Newbury Park.

Turtle, H.R., Croft, W.B., 1991. Evaluation of an inference network-based retrieval model. ACM Transactions on Information Systems 9 (3), 187–222.

van Vuuren, W., 1998. Organisational Failure: An Exploratory Study in the Steel Industry and the Medical Domain. PhD thesis, Technical University of Eindhoiven, Netherlands.